LAMP-TR-089 July 2002
CS-TR-4381
UMIACS-TR-2002-61

The Web as a Parallel Corpus

Philip Resnik, Noah A. Smith

Language and Media Processing Labratory
Instititue for Advanced Computer Studies
College Park, MD 20742

Abstract

Parallel corpora have become an essential resource for work in multi-lingual natural language processing. In this report, we describe our work using the STRAND system for mining parallel text on the World Wide Web, first reviewing the original algorithm and results and then presenting a set of signicant enhancements. These enhancements include the use of supervised learning based on structural features of documents to improve classication performance, a new content-based measure of translational equivalence, and adaptation of the system to take advantage of the In-ternet Archive for mining parallel text from the Web on a large scale. Finally, the value of these techniques is demonstrated in the construction of a significant parallel corpus for a low-density language pair.

***The support of the LAMP Technical Report Series and the partial support of this research by the National Science Foundation under grant EIA0130422 and the Department of Defense under contract MDA9049-C6-1250 is gratefully acknowledged.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUL 2002		**00-07-2002 to 00-07-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The Web as a Parallel Corpus	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Language and Media Processing Laboratory,Institute for Advanced Computer Studies,University of Maryland,College Park,MD,20742-3275**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**30**	

The Web as a Parallel Corpus

Philip Resnik* and Noah A. Smith†

Abstract

Parallel corpora have become an essential resource for work in multilingual natural language processing. In this report, we describe our work using the STRAND system for mining parallel text on the World Wide Web, first reviewing the original algorithm and results and then presenting a set of significant enhancements. These enhancements include the use of supervised learning based on structural features of documents to improve classification performance, a new content-based measure of translational equivalence, and adaptation of the system to take advantage of the Internet Archive for mining parallel text from the Web on a large scale. Finally, the value of these techniques is demonstrated in the construction of a significant parallel corpus for a low-density language pair.

1 Introduction

Parallel corpora — bodies of text in parallel translation, also known as *bitexts* — have taken on an important role in machine translation and multilingual natural language processing. They represent resources for automatic lexical acquisition (e.g. Gale and Church, (1991); Melamed, (1997)), they provide indispensable training data for statistical translation models (e.g. Brown et al., (1990); Melamed (2000)), and they can provide the connection between vocabularies in cross-language information retrieval (e.g. Davis and Dunning (1995); Landauer and Littman (1990); see also Oard (1997)). More recently, researchers at Johns Hopkins University and the University of Maryland have been exploring new ways to exploit parallel corpora in order to develop *monolingual* resources and tools, using a process of annotation, projection, and training: given a parallel corpus of English with a less resource-rich language, we project English annotations across a parallel corpus to a second language, using word-level alignments as the bridge, and then use robust statistical techniques in learning from the resulting noisy annotations (Cabezas, Dorr, and Resnik, 2001; Diab

*Department of Linguistics and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. Email: resnik@umiacs.umd.edu

†Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218. Email: nasmith@cs.jhu.edu

and Resnik, 2002; Hwa et al., 2002; Lopez et al., 2002; Yarowsky, Ngai, and Wicentowski, 2001; Yarowsky and Ngai, 2001; Riloff, Schafer, and Yarowsky, 2002).

For these reasons, parallel corpora can be thought of as a critical resource. Unfortunately, they are not readily available in the necessary quantities. Until very recently, for example, statistical work in machine translation focused heavily on French-English translation because the Canadian parliamentary proceedings (Hansards) in English and French were the only large bitext available. Things have improved somewhat, but it is still fair to say that for all but relatively few language pairs, parallel corpora tend to be accessible only in specialized forms such as United Nations proceedings,[1], religious texts (Resnik, Olsen, and Diab, 1999), localized versions of software manuals (Resnik and Melamed, 1997), and the like. Even for the top handful of majority languages, the available parallel corpora tend to be unbalanced, representing primarily governmental or newswire-style texts. In addition, like other language resources, parallel corpora are often encumbered by fees or licensing restrictions. For all these reasons, it is difficult to follow the "more data are better data" advice of Church and Mercer ((1993)), abandoning balance in favor of volume, with respect to parallel text.

And then there is the Web. People tend to see the Web as a reflection of their own way of viewing the world — as a huge semantic network, or an enormous historical archive, or a grand social experiment. We are no different: as computational linguists working on multilingual issues, we view the Web as a great big body of text waiting to be mined, a huge fabric of linguistic data interwoven with parallel threads.

This report describes our techniques for mining the Web in order to extract the parallel text it contains. It presents, in revised and considerably extended form, our early work on mining the Web for bilingual text (STRAND, (Resnik, 1998; Resnik, 1999)), incorporating new work on content-based detection of translations (Smith, 2001; Smith, 2002) and efficient exploitation of the Internet Archive (Resnik and Smith, 2002). In Section 2 we lay out the STRAND architecture, which is based on the insight that translated Web pages tend quite strongly to exhibit parallel *structure*, permitting their exploitation even without looking at content; we also show how we have improved STRAND's performance by training a supervised classifier using structural parameters rather than relying on manually tuned thresholds. In Section 3 we present an approach to detecting translations that relies entirely on *content* rather than structure, demonstrating performance comparable to STRAND's using this orthogonal source of information. In Section 4 we describe how we have adapted the STRAND approach to the Internet Archive, dramatically improving our ability to identify parallel Web pages on a large scale. Section 5 puts all the pieces together, using structural and combined content-structure matching of pages on the Internet Archive in order to obtain a sizable corpus of English-Arabic Web document

[1] E.g., via LDC, http://www.ldc.upenn.edu.

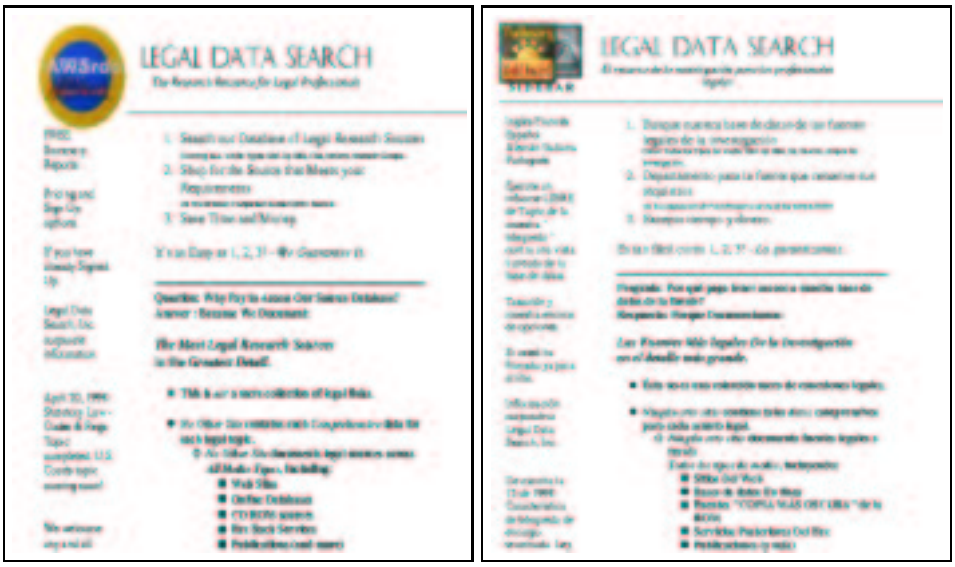

Figure 1: Example of a candidate pair.

pairs. In Section 6 we present our conclusions and thoughts on future work.

2 The STRAND Web-Mining Architecture

STRAND (Resnik, 1998; Resnik, 1999) is an architecture for structural translation recognition, acquiring natural data. Its goal is to identify naturally occurring pairs of Web pages in parallel translation. In order to do this, it exploits an observation about the way that Web page authors disseminate information in multiple languages: when presenting the same *content* in two different languages, authors exhibit a very strong tendency to use the same document *structure* (e.g. Figure 1). STRAND therefore locates pages that *might* be translations of each other, via a number of different strategies, and filters out page pairs where the page structures diverge by too much.

In this section we describe how STRAND works, and we also discuss several related Web mining methods, focusing on the overall architecture these systems have in common and the important system-specific variations. We then show how tuning STRAND's structural parameters using supervised training can significantly increase its performance.

2.1 STRAND

Finding parallel text on the Web consists of three main stages:

- Location of pages that might have parallel translations
- Generation of candidate pairs that might be translations
- Structural filtering-out of non-translation candidate pairs

We consider each of these in turn.

2.1.1 Locating Pages.

The original STRAND architecture accomplished the first step by using the AltaVista search engine (www.av.com) to search for two types of Web pages. A *parent* page is one that contains hypertext links to different-language versions of a document, e.g. if we were looking for English and French bitexts, the page at the left in Figure 2 would lead us to one such candidate pair.

A *sibling* page is a page in one language that itself contains a link to a version of the same page in another language; e.g. the page at the right of Figure 2 contains a link at the top that says "English".

More recent versions of STRAND (unpublished) added a "spider" component for locating pages that might have translations. Given a list of Web sites thought to contain bilingual text for a given language pair, it is possible to download all the pages on the site, any of which might have a translation on that site. Although simple to implement, this method of locating pages shifts the burden of narrowing down the possibilities to the process of generating candidate document pairs.

2.1.2 Generating Candidate Pairs.

Pairing up potentially translated pages is simple when a search engine has been used to generate parent or sibling pages: one simply pairs together the two child pages linked to by the parent, or the sibling page together with the page it links to.

When all the pages on a site are under consideration, the process is rather different. The simplest possibility is to separate the pages on a site into the two languages of interest using automatic language identification (e.g. Dunning (1994)), throwing away any pages that are not in either language, and then generate the cross product. This potentially leads to a very large number of candidate page pairs, and for most of them there is no particular reason to believe they are parallel translations, other than the fact that they appear on the same Web site. The "spider" component of STRAND adds a URL-matching stage, exploiting the fact that the directory structure on many Web sites reflects parallel organization when pages are translations of each other. Matching is done by manually creating a list of substitution rules (e.g. `english` $\Rightarrow$ `big5`), and, for each English URL, applying all possible rules to generate URLs that might appear on the list of pages for the other language. If such a URL

Figure 2: Examples of a parent page and sibling page used to find candidate document pairs.

is found, the pair with similar URLs is added to the list of candidate document pairs. For example, if an English-Chinese site contains a page with URL http://mysite.com/english/home_en.html, one combination of substitutions might produce the URL http://mysite.com/big5/home_ch.html, which are probably worth considering as a likely candidate pair.[2] Owing to the combinatorics (an exponential number of possible substitutions), only a fixed number of substitution combinations could be tried per English URL; however, in Section 4.3 we describe a more scalable URL-matching algorithm.

Another possible criterion for matching is the use of document lengths. Texts that are translations of each other tend to be similar in length, and it is reasonable to assume length(E) $\approx f$(length(F)) where f is a linear function whose parameters are tuned for languages E and F. The use of a document-length filter is described in Smith (2001), where such a filter is shown, at the sentence level, to reduce the size of the search space exponentially in the confidence (p in a $(1-p)$-confidence interval for a linear regression model) with only linear loss of good pairs.

[2] Big5 is the name of a commonly used character encoding for Chinese.

2.1.3 Structural Filtering.

The heart of STRAND is a structural filtering process that relies on analysis of the pages' underlying HTML to determine a set of pair-specific structural values, and then uses those values to decide whether the pages are translations of each other.

The first step in this process is to linearize the HTML structure and ignore the actual linguistic content of the documents. Both documents in the candidate pair are run through a markup analyzer that acts as a transducer, producing a linear sequence containing three kinds of token:

`[START:element_label]`	e.g. `[START:A]`, `[START:LI]`
`[END:element_label]`	e.g. `[END:A]`
`[Chunk:length]`	e.g. `[Chunk:174]`

The second step is to align the linearized sequences using a standard dynamic programming technique (Hunt and McIlroy, 1975). For example, consider two documents that begin as follows:

<HTML>	<HTML>
<TITLE>Emergency Exit</TITLE>	<TITLE>Sortie de Secours</TITLE>
<BODY>	<BODY>
<H1>Emergency Exit</H1>	Si vous êtes assis à
If seated at an exit and	côté d'une...
⋮	⋮

The aligned linearized sequence would be as follows:

`[START:HTML]`	`[START:HTML]`
`[START:TITLE]`	`[START:TITLE]`
`[Chunk:12]`	`[Chunk:15]`
`[END:TITLE]`	`[END:TITLE]`
`[START:BODY]`	`[START:BODY]`
`[START:H1]`	
`[Chunk:12]`	
`[END:H1]`	
`[Chunk:112]`	`[Chunk:122]`

Using this alignment, we compute four scalar values that characterize the quality of the alignment.

- dp The difference percentage, indicating non-shared material, i.e. alignment tokens that are in one linearized file but not the other.
- n The number of aligned non-markup text chunks of unequal length.
- r The correlation of lengths of the aligned non-markup chunks.
- p The significance level of the correlation r.

The difference percentage (dp) quantifies the extent to which there are mismatches in the alignment — sequence tokens on one side that have no corresponding token on the other side. In the example above, one document contains an `H1` header that is missing from the second document. Large numbers of such mismatches can indicate that the two documents do not present the same material to a great enough extent to be considered translations. This can happen, for example, when two documents are translations up to a point, e.g. an introduction, but one document goes on to include a great deal more content than another. Even more frequently, the proportion is high when two documents are *prima facie* bad candidates for a translation pair.

The number of aligned non-markup text chunks (n) helps characterize the quality of the alignment. The dynamic programming algorithm tries to optimize the correspondence of identical tokens, which represent markup.[3] As a side-effect, the non-markup text chunks are placed in correspondence with each other (e.g. the "Emergency Exit" and "Sortie de Secours" chunks in the above example). The more such pairings are found, the more likely the candidate documents are likely to represent a valid translation pair.

The remaining two parameters (r and p) quantify the extent to which the corresponding non-markup chunks are correlated in length. When two documents are aligned with each other and are valid translations, there is a reliably linear relationship in the length of translated chunks of text — short pieces correspond with short pieces, medium with medium, and long with long. The Pearson correlation coefficient r for the lengths will be closer to 1 when the alignment has succeeded in lining up translated pieces of text, and the p value quantifies the reliability of the correlation, e.g. the standard threshold of $p < .05$ indicates 95% confidence that the correlation was not obtained by chance.

In our original work, we used fixed thresholds, determined manually by inspection of development (non-test) data for English-Spanish, to decide whether a candidate pair should be kept or filtered out. Thresholds of $dp < 20\%$ and $p < 0.05$ were used.

2.2 STRAND Results

Like most search tasks, performance at finding parallel Web pages can be evaluated using standard measures of precision and recall and by combining those figures using the F-measure. It is not possible for us to measure recall relative to the entire set of document pairs that should have been found — this would require exhaustive evaluation using the entire Web, or pooling of results from a large number of different systems as done in the TREC information retrieval evaluations. Therefore, recall in this setting is measured relative to the set of candidate pairs that was generated.

[3] "Non-markup" tokens with exactly the same length almost always turn out to be pieces of identical markup, e.g. `key=value` pairs within HTML tags.

Since the "truth" in this task is a matter for human judgment, we rely on bilingual speakers to independently judge whether page pairs are actually translations of each other for any given test set. In our experience *no* bilingual speaker is 100% comfortable saying that another person's translation is a good translation, so in creating the gold standard we instead ask, "Was this pair of pages intended to provide the same content in the two different languages?" Asking the question in this way leads to high rates of inter-judge agreement, as measured using Cohen's κ measure.

2.2.1 Using Manually-Set Parameters.

Using the manually set thresholds for *dp* and *n*, we have obtained 100% precision and 64.1% recall (extrapolated to include a set of unjudged items) in an experiment using STRAND to find English-French Web pages (Resnik, 1999). A modified version of STRAND was used to obtain English-Chinese pairs (see related work, below), and in a similar formal evaluation, we found that the resulting set had 98% precision and 61% recall for Chinese.[4] Both these results are consistent with our preliminary findings for English-Spanish using a less rigorous evaluation (using the judgments of the first author rather than independent bilingual evaluators) and a very small test set, where precision was near ceiling and recall was in the vicinity of 60% (Resnik, 1998).

2.2.2 Optimizing Parameters using Machine Learning.

Based on experiments with several language pairs, it appears that STRAND's structure-based filter consistently throws out a bit more than one third of the candidate document pairs it has found in order to maintain its precision in the 98-100% range. It does so by respecting parameter thresholds that were determined manually using English-Spanish development data; the same parameters seem to have worked reasonably well not only for the English-Spanish, but also for English-French and English-Chinese pairs. It is possible, however, that classification can be tuned for better performance. In order to investigate this possibility, we took a machine-learning approach: we used the four structural values (*dp*, *n*, *r*, and *p*) as features characterizing each document pair, and treated the problem as a binary decision task, using supervised learning to make an attempt at better predicting human judgments.

Using the English-French data, we constructed a threefold cross-validation experiment using decision tree induction to predict the class assigned by the human judges. The decision tree software was C5.0.[5] Each fold had 87 test items and 174 training items. Precision and recall computations include extrapolation from the judged set to an additional 16,328 unjudged items; the results are

[4] http://umiacs.umd.edu/~resnik/strand/

[5] Available at http://www.rulequest.com/demoeula.html.

	precision	recall
Untuned	1.000	0.641
Fold 1	0.913	0.834
Fold 2	1.000	0.943
Fold 3	1.000	0.724
Average	0.971	0.834

Table 1: Effects of parameter tuning.

reported in Table 1 together with baseline results from STRAND's untuned classifier as reported above.

Without tuning, the manually-set parameters result in good document pairs being discarded 36% of the time. Our cross-validation results indicate that tuning the parameters cuts that figure by more than half: only 17% of the good pairs will be discarded, at a cost of admitting 3 false positives from every 100 candidates pairs.

2.3 Related Work

Several other systems for discovering parallel text, developed independently, can be described as operating within the same three-stage framework as STRAND.

Parallel Text Miner (PTMiner, Chen and Nie (2000)) exploits already-existing Web search engines to locate pages by querying for pages in a given language that contain links to pages that are likely to be in the other language of interest. Once bilingual sites are located, they are crawled exhaustively. In order to generate candidate pairs, PTMiner uses a URL-matching process similar to the one described above; for example, the French translation of a URL like http://www.foo.ca/english-index.html might be http://www.foo.ca/french-index.html. Their matching process uses a mapping of language-specific prefixes and suffixes, and does not handle the case where URL-matching requires multiple substitutions. PTMiner also applies a length filter and automatic language identification to verify that the pages are in the appropriate languages. Chen and Nie report a 95%-precise English-French corpus of 118MB/135MB of text and a 90%-precise English-Chinese corpus of 137MB/117MB of text, based on inspection.

Bilingual Internet Text Search (BITS, Ma and Liberman (1999)) starts with a given list of domains to search for parallel text. It operates by sampling pages from each domain and identifying their languages; if a domain is deemed to be multilingual, all pages on the site are crawled exhaustively. BITS appears to consider all possible combinations of Web page pairs in the two languages (i.e. the full cross product within each site), and filters out bad pairs by using a large

bilingual dictionary to compute a content-based similarity score and comparing that score to a threshold. For each page pair, the similarity score is

$$similarity(A, B) = \frac{\text{\# translation token pairs}}{\text{\# tokens in A}} \quad (1)$$

Translation token pairs are considered within a fixed window (i.e., a distance-based measure of cooccurrence is used).[6] In addition to cross-lingual lexical matching, BITS filters out candidate pairs that do not match well in terms of file size, anchors (numbers, acronyms, and some named entities), or paragraph counts. Using an English-German bilingual lexicon of 117,793 entries, Ma and Liberman report 99.1% precision and 97.1% recall on a hand-picked set of 600 documents (half in each language) containing 240 translation pairs (as judged by humans). This technique yielded a 63MB parallel corpus of English-German.

Other work on Web mining has been done by Jinxi Xu of BBN (personal communication), who began with our STRAND implementation and added a module for automatically learning string-substitution patterns for URLs, and also implemented a different dynamic programming algorithm for assessing structural alignment. Xu used the modified STRAND to obtain 3376 Chinese-English document pairs, which we evaluated formally (see above), determining that the set has 98% precision and 61% recall.

In addition, STRAND has been re-implemented by David Martinez and colleagues at Informatika Fakultatea in the Basque Country (personal communication), in order to perform exploratory experiments for discovering English-Basque document pairs.

It is worth noting that STRAND, PTMiner, and BITS are all largely independent of linguistic knowledge about the particular languages, and therefore very easily ported to new language pairs. With the exception of the use of a bilingual dictionary in BITS, these systems require, at most, a set of URL substring patterns for the URL pattern-matching stage (e.g. $big5 \sim english$ in the example above; see further discussion in Section 4.3), and a modest amount of monolingual data for training n-gram based language identifiers (typically 50,000 to 100,000 characters of text per language).

Word-level translations are worth exploiting when they are available, as was the case for the BITS system. In Section 3 we describe a bitext-matching process using a content-based similarity score grounded in information theory, and in Section 5 we show how structural and content-based criteria can be combined in order to obtain performance superior to that obtained by either method alone.

[6] Many details of this technique are left unspecified in (Ma and Liberman, 1999), such as the threshold for the similarity score, the distance threshold, and matching of non-one-word-to-one word entries in the dictionary.

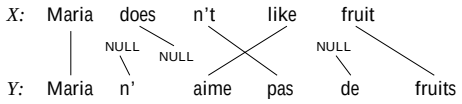

Figure 3: An example of two texts with links shown. There are seven link tokens, five of which are lexical (non-NULL) in X (the English side), six in Y (French).

3 Content-Based Matching

The approach discussed thus far relies heavily on document structure. However, as Ma and Liberman (1999) point out, not all translators create translated pages that look like the original page. Moreover, structure-based matching is applicable only in corpora that include markup, and there are certainly multilingual collections on the Web and elsewhere that contain parallel text without structural tags. Finally, other applications for translation detection exist, such as sub-document text alignment and cross-lingual duplicate detection (i.e., location of already-existing translations in a multilingual corpus). All these considerations motivate an approach to matching of translations that pays attention to similarity of content, whether or not similarities of structure exist.

We present here a generic score of translational similarity that is based upon any word-to-word translation lexicon (hand-crafted or automatically generated, or a combination, and possibly highly noisy). The technique is shown to perform competitively with the structure-based approach of STRAND on the task of identifying English-French document translations.

3.1 Quantifying Translational Similarity

We define a cross-language similarity score, *tsim*, for two texts by starting with generative, symmetric word-to-word model of parallel texts (Melamed's Method A (2000)).[7] Let a *link* be a pair (x, y) where x is a word in language L_1 and y is a word in L_2. The model consists of a bilingual dictionary that gives a probability distribution p over all possible link types. Within a link, one of the words may be NULL, but not both. In the generative process, a sequence of independent link tokens is sampled from that distribution. The model does not account for word order. An example of two texts with links is illustrated in Figure 3.

Next, we desire to compute the probability of the most probable link sequence that could have accounted for the two texts.[8] The probability of a

[7] We use the term "text" to refer to a sequence of words of any length.

[8] Of course, all permutations of a given link sequence will have the same probability (since the links are sampled independently from the same distribution), so the order of the sequence

link sequence is simply the product of the p probabilities of the links it contains. As noted by Melamed (2000), this problem of finding the best set of links is the maximum-weighted bipartite matching problem (MWBM): Given a weighted bipartite graph $G = (V_1 \cup V_2, E)$ with edge weights $c_{i,j} (i \in V_1, j \in V_2)$, find a matching $M \subseteq E$ such that each vertex has at most one edge in M, and $\sum_{e \in M} c_{i,j}$ is maximized. The fastest known MWBM algorithm runs in $O(ve + v^2 \log v)$ time (Ahuja, Magnati, and Orlin, 1993). Applied to this problem, that is $O(\max(|X|, |Y|)^3)$.

To use MWBM to find the most probable link sequence, let the L_1 words be V_1 and the L_2 words be V_2. If two words x, y have $p(x, y) > 0$, an edge exists between them with weight $\log p(x, y)$. Hence a sum of weights of links in a matching will be the log-probability of the (unordered) link sequence, and maximizing that sum maximizes the probability.

The similarity score should be high when many of the link tokens in the best sequence do *not* involve NULL tokens. Further, it should normalize for text length. Specifically, the score is:

$$tsim = \frac{\log \Pr(\text{two-word links in best matching})}{\log \Pr(\text{all links in best matching})} \tag{2}$$

This score is an application of Lin's (1998) information theoretic definition of similarity. Starting with a set of axioms, Lin derives the measure

$$sim(X, Y) = \frac{\log \Pr(\text{common(X, Y)})}{\log \Pr(\text{description(X, Y)})} \tag{3}$$

where X and Y are any objects generated by a probabilistic model.

In this discussion, we seek to show how multiple linguistic resources can be exploited together to recognize translation. Therefore, the measure is simplified by assuming that all links in a given translation lexicon are equiprobable. This reduces the formula for *tsim* to

$$tsim = \frac{\#\text{ two-word links in best matching}}{\#\text{ links in best matching}} \tag{4}$$

Another reason to compute *tsim* under the equiprobability assumption is that we need not compute the MWBM, but only find the maximum cardinality bipartite matching (MCBM), since all potential links have the same weight. An $O(e\sqrt{v})$ (or $O(|X| \cdot |Y| \cdot \sqrt{|X| + |Y|})$ for this purpose) algorithm exists for MCBM (Ahuja, Magnati, and Orlin, 1993). For example, if the matching shown in Figure 3 is the MCBM (for some translation lexicon), then $tsim(X, Y) = \frac{4}{7}$ under the simplifying assumption.

Melamed (2000) used a greedy approximation to MWBM called competitive linking. Competitive linking iteratively selects the edge with the highest

is not important.

weight, links those two vertices, then removes them from the graph. (Ties are broken at random.) A heap-based implementation of competitive linking runs in $O(\max(|X|,|Y|)\log\max(|X|,|Y|))$. Under the equiprobability assumption, all the weights are the same, so that competitive linking proceeds simply by randomly making links until no more can be made.

If definition (4) is applied to pairs of documents in the *same* language, with a "translation lexicon" defined by the identity relation, then *tsim* is a variant of resemblance (r), as defined by Broder et al. (1997) for the problem of monolingual duplicate detection, except that *tsim* has the advantage of being token-based rather than type-based, incorporating word frequency. The key result is that *any* translation lexicon (or, importantly, union thereof) containing a set of word-to-word entries can be applied to the computation of *tsim*.

We have demonstrated that this score can be used to extract translationally equivalent English-Chinese sentence pairs from even a noisy space with high precision (Smith, 2002). It was also shown that combining multiple sources of word-level translation information (dictionaries, word-to-word translation models, cognates) had positive effects on performance on the sentence-matching task. The competitive linking approximation was also shown to perform nearly as well as MCBM.

This technique of using a translation model to define translational similarity is generic to different sources of lexical translation information. An important feature is that it can be used with *any* symmetric translation model where events can be divided into those which both sides of a bitext have in common and those which affect only one side.

3.2 Experiment

We now apply our content-based similarity measure to the candidate pair classification task presented by STRAND. Recall that both the original STRAND classifier and those learned using decision tree methods, described in Section 2.2.2, use only structural features of the documents to determine whether they are translations. Here we apply the *tsim* score to the same task and compare the results with those of the original STRAND classifier.

3.2.1 Translation Lexicon.

The word-level translation lexicon is derived from several sources.

The first is an English-French dictionary (a total of 34,808 entries, 4,021 of which are not one-to-one).[9] Each n-to-m entry was processed by stoplisting and then extracting all word-pairs in the remaining cross-product. This technique is liable to introduce noise to the dictionary, so we used only cross-products that

[9] This dictionary was generated by Gina Levow, who kindly made it available to us, using a dictionary derived from one available at http://www.freedict.com. It contains morphological variants but does not include character accents.

generated four new entries or less. The result is 39,348 word pairs, 9,045 of which contain two words present in the corpora.

A word-to-word translation model (Melamed, 2000) was trained on a verse-aligned Bible (15,548 verses, averaging 25.5 English words, 23.4 French words after tokenization). Instead of using competitive linking, we used the maximum weighted bipartite matching algorithm in estimating the parameters of this model. The result includes all word pairs with positive probability (though the probabilities are subsequently ignored) and contains 13,762 word pairs.

The third source comprises English-French cognate pairs, identified using the method of Tiedemann (1999). Tiedemann's approach involved learning language specific character-to-character weights for the computation of weighted edit-distance to measure cognate-ness. He used a list of known cognates to train the weights. We instead used the weighted translation pairs in the translation model lexicon. Hence the resources required to extract cognates in this way are no different from those required for the translation model. The result is 35,513 word pairs. An additional set of 11,264 exact string matches were added. These entries were undoubtedly quite noisy.

The union of these translation lexicons consists of 68,003 unique word pairs. The experiment used only this union translation lexicon; note that the entries in this lexicon are not weighted (they are assumed to be equiprobable).

3.2.2 Results.

In order to compare *tsim* with structural similarity scoring, we applied it to 325 English-French web-document pairs for which human evaluations were carried out in Section 2.[10] As there is only one feature under consideration (*tsim*), the classifier must be a threshold on that value. At different thresholds, Cohen's κ score of agreement (with each of Resnik's (1999) two judges and their intersection) may be computed for comparison with STRAND, along with recall and precision against a gold standard (for which we use the intersection of the judges — the set of examples where the judges agreed).

Computing *tsim* (MCBM on the words in the document pair) is not tractable for very large documents and translation lexicons. However, in preliminary comparisons, we found that representing long documents by as few as their first 500 words results in excellent performance on the κ measure. This allows $O(1)$ estimation of *tsim* for two documents. Further, the competitive linking algorithm appears to be as reliable as MCBM, and it runs significantly faster in practice. The results reported here approximated *tsim* in this way.

Of the 325 pairs, 32 were randomly selected as a development set. We manually selected a threshold of $\tau = 0.15$. This value was chosen because it

[10] One additional pair was thrown out because it contained compressed data; it is assumed that pair would not pass a language identification filter.

Comparison	N	Pr(Agree)	κ	*prec*	*rec*	F
J1, J2	245	0.98	0.96			
J1, STRAND	250	0.88	0.70			
J2, STRAND	284	0.88	0.69			
J1 $\cap$ J2, STRAND	241	0.90	0.75	**0.963**	0.684	0.800
J1, *tsim*($\tau = 0.15$)	249	0.92	0.83			
J2, *tsim*($\tau = 0.15$)	283	0.92	0.82			
J1 $\cap$ J2, *tsim*($\tau = 0.15$)	240	0.93	**0.85**	0.680	**0.921**	0.782

Table 2: Comparison with STRAND. The test set is 293 of the 326 pairs in Resnik's (1999) test set. The 32 development pairs were used to manually select the 0.15 threshold. N is the number of examples for which judgement-comparison was possible in each case (human judges were sometimes undecided; those cases are ignored in computing κ).

was the one that resulted in the maximum κ score on the development set.[11] κ scores against each judge and their intersection were then computed at that threshold on the test set (the remaining 293 pairs). These are compared to κ scores of the STRAND system, on the same test set, in Table 2. In every case, the *tsim* classifier agreed more strongly with the human evaluations, and its F score is competitive with that of STRAND. A simple combination of the classifiers by disjunction (i.e., "(X, Y) is a translation pair if either classifier says so") yields precision 0.768, recall 0.961, $F = 0.854$, and κ (with the judges' intersection) at 0.878.

At $\tau = 0.15$, precision was 0.680 and recall was 0.921, $F = 0.782$. On the same set, STRAND structural classification achieved 0.963 precision and 0.684 recall, $F = 0.800$. Figure 4 shows κ, precision, and recall each plotted against τ.

For this application, the structural and content-based classifiers have complementary strengths; the former is highly precise while the latter gives high recall. The content-based classifier also more reliably predicts human judgements (based on Cohen's κ).

Given two high-performing methods that use orthogonal information for identifying good candidate pairs (one using only structure, the other using only content), the natural question is whether the techniques can be combined for even better performance. We answer this question in affirmative in Section 5. First, however, we describe how we have adapted the STRAND architecture to the Internet Archive in order to generate the candidate pairs on a scale that was previously unattainable.

[11] One could select such a threshold to maximize any objective function over the development set.

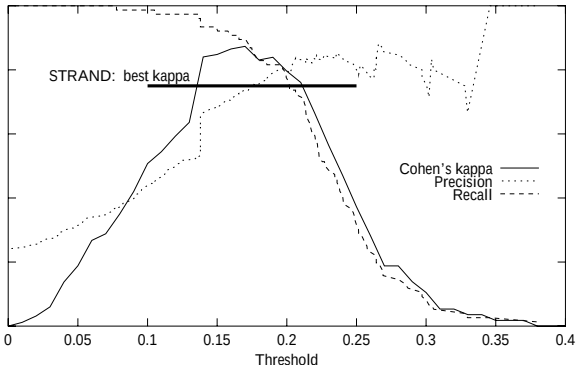

Figure 4: Performance measures as the threshold varies (all measures are on the test set): the κ agreement score with the two judges' intersection, precision, and recall. The κ score obtained by STRAND is shown as well.

4 Exploiting the Internet Archive

One of the difficulties with doing research on Web mining is that large-scale crawling of the Web is a significant enterprise. Chen and Nie's (2000) PTMiner represents one solution, a carefully thought-out architecture for mining on a large scale. Here we present a different solution, taking advantage of an existing large-scale repository of Web pages maintained on an ongoing basis by an organization known as the Internet Archive.

4.1 The Internet Archive

The Internet Archive is a non-profit organization attempting to archive the entire publicly available Web, preserving the content and providing free access to researchers, historians, scholars, and the general public.[12] Data come from crawls done by Alexa Internet, and hence they represent an industry-level resource of the sort not easily constructed within academia. At present, the Archive contains 120T (terabytes) of data, by a conservative estimate, and it is growing at approximately 8T per month. Text on the archive comprises over 10 billion Web pages, and the estimated duplicate rate is a factor of 2, i.e. two copies of everything.[13]

The Internet Archive provides public access to the data via the "Wayback Machine" Web interface. As of this writing, a search for the ACL home page

[12] See http://www.archive.org.

[13] We are grateful to Paula Keezer of Alexa for these figures.

brings up links to 27 snapshots of that page dating back to June 7, 1997.[14] The reader can get to that page directly on the Wayback Machine using a URL that points to the Internet Archive and provides both the desired page and the timestamp indicating which snapshot to retrieve.[15]

The Archive also provides researchers direct access to its data via accounts on their cluster. The data are stored on the disk drives of approximately 300 machines, each running some variety of Unix, creating what is in essence one huge file system. This provides a researcher the remarkable sensation of having the entire Web on his or her hard drive.

4.2 Properties of the Archive

Mining terabytes on the Archive presents a number of challenges.

- The Archive is a temporal database, but it is not stored in temporal order. Hence a document and its translation may be in files on different machines; a global merge of data is required for any hope of complete extraction.
- Extracting a document for inspection is an expensive operation involving text decompression.
- The Archive's size makes it essential to keep computational complexity low.

On the other hand, aspects of the Archive's architecture turned out to make rapid development remarkably feasible.

- Almost all data are stored in compressed plain text files, rather than in databases.
- The data relevant for our purposes are organized into archive files (arcfiles), which containing the stored pages, and index files, which contain plain text tuples $\langle \text{URL}, \text{timestamp}, \text{arcfile}, \text{offset}, \ldots \rangle$.
- A suite of tools exists for, e.g., extracting individual pages from archive files.
- The Archive's infrastructure for cluster computing makes it easy to write Unix scripts or programs and run them in parallel across machines.

The last of these, the cluster computing tools, turned out to drastically reduce the time needed to port STRAND to the Archive, and literally halved the size of the STRAND code base. The centerpiece in Archive cluster computing is a parallelization tool called `p2`, which offers a UNIX command-line interface

[14] http://www.cs.columbia.edu/~acl/home.html

[15] http://web.archive.org/web/19970607032410/http://www.cs.columbia.edu/~acl/home.html. This is a single big URL.

that allows one to specify (a) a parallelizable task, (b) a way to split it up, (c) a way to combine the results, and (d) a set of processors among which to divide the task. The `p2` tool divides up tasks intelligently, invoking each parallel computation on the local machine where the data reside.[16]

4.3 STRAND on the Archive

In adapting STRAND's three-stage process to the Internet Archive, the primary challenge was in the first two steps, locating possible translations and matching them up to produce candidate document pairs. Structural filtering remained essentially unchanged.

Generating candidate pairs on the Archive involves the following steps:

1. Extracting URLs from index files using simple pattern matching
2. Combining the results from Step 1 into a single huge list
3. Grouping URLs into buckets by handles
4. Generating candidate pairs from buckets

Steps 1 and 2 are done via a parallel search operation plus combination of results, e.g. extracting URLs (and their associated bookkeeping data) using a pattern like `/(.hk|.tw|.cn)/` to extract all URLs in the Hong Kong, Taiwan, or China domains.[17]

Step 3 is potentially tricky owing to computational complexity issues. As noted in Section 2.1.2, examining the cross product of a site's page sets in two different languages is potentially very expensive, and matching documents by similarity of URLs can represent a combinatoric process in the general case.

We arrived at an algorithmically simple solution that avoids this problem, but is still based on the idea of language specific substrings (LSSs). The idea is to identify a set of language-specific URL substrings that pertain to the two languages of interest, e.g. based on language names, countries, character code-sets labels, abbreviations, etc. For example, a set of LSSs for English-Arabic might be as follows:

> 1256, 437, 864, 8859-1, 8859-6, a, ar, ara, arab, arabic, cp1256, cp437, cp864, e, en, eng, english, gb, iso, iso-8859-1, iso-8859-6, latin, latin-1, latin1, uk, us, usa

For each URL, we form a "handle" by subtracting out any substrings that match (insensitive to case) any item on the LSS pattern list. The subtraction process is

[16] The Archive intends to release these cluster tools under the GNU Public License.

[17] We also take advantage of the Archive's list of `.com` domains paired with the nation in which each is registered, making it possible to include commercial sites in the search without an explosion of irrelevant possibilities.

URLs:

```
     saudifrenchbank.com.sa/English/English.htm
 →   saudifrenchbank.com.sa/English/English.htm
```

patterns removed:

```
      a e a a english english

     saudifrenchbank.com.sa/Arabic/arabic.htm
 →   saudifrenchbank.com.sa/Arabic/arabic.htm
```

patterns removed:

```
      a e a a arabic arabic
```

Same handle for both URLs:

```
     sudifrchbnk.com.s//.htm
```

Figure 5: Example of LSS substraction.

implemented reasonably efficiently: if there are p patterns with maximum length l, and the URL's length in characters is u, then the current implementation will do at most $p \times u$ string matches of length no more than l. (Currently we use C `strcmp` for string match.) In practice, this is extremely fast — we can generate handles for nearly 5000 URLs per second on a 6-year-old Sun Ultra 1 workstation.[18]

Figure 5 illustrates handle generation on two real URLs. As one would hope, these two URLs produce the same handle, and as a result, they wind up in the same bucket in Step 3.[19]

In Step 4, the URLs in each bucket are used to generate candidate pairs by taking the cross product and keeping those URL pairs for which the URL bookkeeping data indicates pages that are in the correct languages. For example, given the bucket containing the two URLs in Figure 5, this step would generate a single pair consisting of the URL for the English page and the URL for the Arabic page, assuming the language ID information associated with each URL confirmed it was in the proper language.[20]

At this point, the candidate generation process is complete. The final step is to apply STRAND's structural filtering step to each candidate pair — an operation that can itself be parallelized since each candidate pair can be processed independently.

It is interesting to note that by taking advantage of the Archive's p2 clus-

[18] We are grateful to Bill Pugh of the University of Maryland for suggesting this algorithm.

[19] Conceptually, they hash to the same bucket in a hash table; in practice on the Archive it turns out to be more efficient to create buckets by doing a parallel sort of the entire URL set using the handle as the key, and then creating buckets based on identical handles being on adjacent lines.

[20] The Internet Archive tags its data for language using standard n-gram language identification techniques.

ter computing tool, together with its simple flat-text representations, adapting STRAND's candidate generation process resulted in a dramatic reduction in the size of the program, cutting it in half, as measured in lines of code.

5 Building an English-Arabic Corpus

In the previous sections, we have described methods and results for structural matching, content-based matching, and for dramatically scaling up the number of candidate pairs that can be generated for any given language pair by using the industrial-strength Web crawls stored on the Internet Archive. In this section we put all these pieces together, describing an experiment in mining the Internet Archive to find English-Arabic parallel text. This language pair is of particular global importance, but resources, particularly bilingual text, are not easy to obtain. Moreover, Arabic text is far behind on the Web's exponential growth curve — Arabic text (as opposed to images) did not really start emerging on the Web until the release of Microsoft Windows 98™, which provided Arabic support in its version of Internet Explorer.

5.1 Finding English-Arabic Candidate Pairs on the Internet Archive

The input resources for this search were a list of Internet domains likely to contain Arabic text.[21] The list included twenty-four top-level national domains for countries where Arabic is spoken by a significant portion of the population, including Egypt (`.eg`), Saudi Arabia (`.sa`), Kuwait (`.kw`), etc. In addition, we used a list of `.com` domains known to originate in Arabic-speaking countries. This list provided an additional twenty-one specific domains (e.g., emirates-bank.com, checkpoint.com) — note that it is by no means exhaustive.

In the experiments we report here, we mined two crawls from 2001, comprising 8T and 12T — i.e. less than one sixth of the Archive as it exists today – spread over 27 machines. Our list of URLs with relevant domains, obtained by pattern-matching in Archive index files, numbers 19,917,923 pages.[22] The language-specific substrings given earlier were subtracted from these URLs to generate handles, resulting in 786,880 buckets with an average of twenty-five pages per bucket. When all possible English-Arabic page pairs were generated from all buckets, the result was 8,294 candidate pairs.

A random sample of two hundred candidate pairs was given to two human evaluators, bilingual in English and Arabic, who were asked (independently) to answer the question, for each pair, "Is this pair of pages intended to show the same material to two different users, one a reader of English and the other

[21] We are grateful to Nizar Habash for constructing this list.

[22] Pages with the same URL but different timestamps are counted separately; there were 10,701,622 unique URLs.

	precision	recall
Fold 1	0.9111	0.9229
Fold 2	0.9302	0.9351
Fold 3	0.9565	0.8818
Average	0.9326	0.9133

Table 3: English-Arabic structural classification results. Precision and recall are extrapolated to the entire set of 8,294 candidate pairs.

a reader of Arabic?" The judges' answers showed a Cohen's κ agreement of 0.6955, which is generally considered fair to good reliability. (Qualitatively, one judge was rather more strict than the other; when the stricter judge identified a page pair as valid translations, the less strict judge virtually always agreed.)

5.2 Evaluating Structure-Based Matching

Taking the set of 149 labeled pairs on which the two judges agreed (134 were marked good, 15 bad), we carried out an evaluation of the full candidate set similar to the one for English-French discussed in Section 2.2.2. As before, this was a threefold cross-validation experiment in which decision tree classifiers were tuned on the features extracted for each each candidate pair by structure-based classification. In addition to the four structural scores, we included two language identification confidence scores (one for the English page, one for the Arabic page) — these were available as part of the Internet Archive's bookkeeping information for each URL and required no additional computation on our part. Table 3 shows extrapolated precision and recall of each fold's classifier applied to the entire set of 8,294 candidate page pairs.

The value of the parameter-tuning process is dramatically confirmed by comparing the learned parameters with STRAND's default parameters (manually determined by Resnik (1999)). Precision figures for the three folds were similar on average but far less consistent (1.0 for the first two folds, 0.6667 for the third), and none of the three folds achieved recall above 0.12.

Upon inspection, we discovered that nearly 5,000 of the pairs in our candidate set were from a single domain, maktoob.com. This site supports an online marketplace, and many of the pages discovered by our search were dedicated to specific merchandise categories within that service; a large portion of these were simply "no items available" and one or two similar messages. We ignored this domain completely in order to be conservative about the yield of page pairs, though we note that many of the pages within it are legitimate parallel text that could be extracted if a duplicates filter were applied).[23]

[23] One of our human evaluators confirmed that no other domains appeared to significantly

In order to construct a final classifier, we trained a decision tree on all 200 of our manually judged examples. This was then applied to the candidate pairs, producing a set of 1,741 HTML document pairs judged to be valid translations of each other. Converting from HTML to plain text and tokenizing, the English documents in this corpus total approximately 818,102 tokens, with an average of 470 tokens per document; the Arabic side contains 1,025,461 tokens, averaging 569 tokens per document.[24]

5.3 Combining Structural and Content-Based Matching

We combined the structural and content-based approaches to detecting translations by adding the *tsim* score to the set of structural features associated with each candidate pair, and then training a new decision tree classifier.[25]

Because Arabic is a highly inflected language with many surface forms, we found it necessary to use morphological preprocessing in order to make effective use of a dictionary. For English, we tokenized the text and used the WordNet lemmatizer to strip suffixes. The Arabic texts were tokenized at punctuation, then Romanized and converted to root forms using a morphological analysis tool (Darwish, 2002). This approximately halved the vocabulary size for the Arabic texts (from 89,047 types to 48,212 types).

The translation lexicon used to compute *tsim* contained of 52,211 entries, each containing one English lemma and one Arabic root. Of these, 16,944 contained two items that were both present in the candidate set of 8,295 Web page pairs.[26] The approximations discussed in Section 3.2.2 were used: competitive linking on the first 500 words in each document was used to compute the score.

Carrying out the same cross-validation experiment (on the same random split of data), the combined structural and content-based classifier produced the results in Table 4; these are extrapolated to the full set of 8,294 candidate pairs. Averaged over three folds, the classifier achieved 95.06% precision and 98.95% recall (1.8% and 7.62% better than without *tsim*, respectively). We repeat the average using only STRAND's structural features for reference.

---

dominate the candidate pairs as did maktoob.com, providing some assurance that the rest of the data are diverse.

[24] We converted HTML to text using the `lynx` browser, performed cleanups such as removing references, and tokenized using the tokenizers included with the Egypt statistical MT package (Al-Onaizan et al., 1999). That tokenizer is somewhat aggressive about separating out punctuation, so, being aggressive in the opposite direction, we also tried counting only tokens containing at least one of [A-Za-z] (which excludes punctuation as well as dates, percentages, etc.). Using that very conservative counting method, the size of the English side is approximately 612,673 words. Counting only tokens on the Arabic side which contained at least one non-numeric, non-punctuation character yielded 914,815 words.

[25] We also did perform the experiment using content-based matching alone. However, as noted earlier, *tsim*'s strength is recall, and on this task baseline performance (marking all candidate pairs as good translations) gives 90% precision. Not surprisingly, the performance of the *tsim*-only classifier did not exceed the baseline.

[26] This translation lexicon was used with the kind permission of Kareem Darwish.

	precision	recall
Average (structure only)	0.9326	0.9133
Fold 1	0.9167	1.0000
Fold 2	0.9767	0.9686
Fold 3	0.9583	1.0000
Average	0.9506	0.9895

Table 4: English-Arabic combined structural/content-based classification results. Precision and recall are extrapolated to the entire set of 8,294 candidate pairs.

Tokenization method	English tokens	Arabic tokens
English lemmas, Arabic roots	1,097,674	1,107,600
Egypt tokenizers (Al-Onaizan et al., 1999)	1,170,360	1,512,959
Egypt tokenizers, ignoring words without letters	922,541	1,388,953

Table 5: Yield: the English-Arabic Internet Archive corpus, tokenized several ways

After building a single classifier on all 149 test pairs (the set on which both human judges agreed), we re-classified the entire candidate set. Ignoring again pages from the maktoob.com domain, 2,190 pairs were marked as translations. Table 5 shows word counts for various tokenization schemes: the morphological analysis used for computing *tsim*, the Egypt tokenizer (which is highly aggressive), and counting only tokens with some alphabetic character from the Egypt tokenizer (a conservative approximation).

To summarize the results, using the content-based similarity score as a feature not only improved precision, it increased the size of the corpus (in words) by 43–52%, depending on the tokenization scheme.

To our knowledge, no bilingual text corpus of this magnitude is yet available to the general research community for this language pair. Even if such resources existed for this particular pair, our success with English-Arabic represents a proof of concept for locating bilingual text for other poorly represented language pairs, especially considering the fact that only a small portion of the Internet Archive was searched on this run. Moreover, it bodes well for locating very large quantities of Web text for English paired with better-represented languages.

6 Conclusions

Although efforts at discovering parallel text on the Web were first reported in 1998, Web-based parallel corpora appear to have had only a limited impact on the community. Three reasons for this suggest themselves.

Too few languages. Parallel text from the Web has been made available to the community in only a few pairs of languages. As of this writing, the STRAND Web site,[27] containing URL pairs discovered via STRAND runs, contains collections only for English-French, English-Chinese, and English-Basque, and we are not aware of any other efforts to publicly disseminate Web-based parallel data. Up to this point, it simply has not been easy to search the Web for parallel text in new language pairs. The most difficult part is finding the candidates — a year or two ago, we attempted to apply the original Web-based STRAND to the problem of finding English-Arabic text and we simply were unable to locate enough search engine hits or sites to yield useful results.

Too little data. Very large Web-based parallel text collections are not available to the community. The largest appear to have been obtained by Chen and Nie (2000), who acquired collections for English-French and English-Chinese on the order of 15,000 document pairs using the PTMiner system. However, these collections have not been made available. In contrast, the STRAND collections, which are available to the community in the form of URL pairs, are modest in size: the English-Chinese collection contains fewer than 3500 document pairs, and English-French fewer than 2500.

Difficulty with dissemination. Web-based collections are difficult to distribute. Standard mechanisms of the sort used by LDC — a CD or downloadable file — are fraught with difficult legal issues, since, technically speaking, redistributing the actual content of Web pages could require permission from the author of every page. For example, presumably as a risk reduction strategy, the Web track for TREC-2002 (Text REtrieval Conference) limits its attention to the `.gov` domain and requires the recipient of the data to sign a form that reduces the distributor's liability.[28] Similarly, the Google Programming Contest dataset arrives with a limited-use license, indemnification from third-party claims, and a collection limited to the `.edu` domain, from which, presumably, authors are less likely to bring expensive lawsuits.[29]

A possible fourth reason may have to do with questions about the utility of the data. For example, a Web-based parallel collection may be unpredictable in terms of its coverage and the community is well aware of the dangers of using training data that is not representative of the test domain. A solution to this problem might be to extract topically relevant subsets of the collection

[27] http://umiacs.umd.edu/~resnik/strand/

[28] "The limitation on permitted use ... is intended to reduce the risk of any action being brought by copyright owners, but if this happens the Organisation [recipient] agrees to bear all associated liability" (http://www.ted.cmis.csiro.au/TRECWeb/ access_to_data.html).

[29] http://www.google.com/programming-contest/

for particular domains or applications, but of course this requires a "more is better" approach in order to obtain subsets that are large enough to be useful.

The work reported in this report addresses each of these major problems.

With respect to the number of language pairs, the Internet Archive offers us the largest possible sample of pages on the Web, and our techniques make it easy to explore that collection in an efficient way. Although it is probably impossible to crawl more than a small fraction of the Web, the Internet Archive is storing the results of commercial-scale Web crawling and has as its explicit mission the permanent storage of everything that can be found. The fact that we were able to find a substantial quantity of English-Arabic text — on the order of a million words per side, looking at less than sixth of today's Archive — offers the hope that it will be possible to find data for the less well-represented language pairs, if and when those data actually exist. Moreover, the final implementation we described here retains the almost entirely language-independent character of the original STRAND system, adding only the requirement of a reasonable translation lexicon. Therefore success in mining for parallel text in other languages depends primarily on whether the data exist on the Archive.

With regard to corpus size, we demonstrated that the recall of structural matching, and hence its yield, can be significantly improved by simple and automatic classifier construction, requiring only a few hours' work from a bilingual annotator to create the training material. These results are further improved by adding content-based similarity as a feature. The success with English-Arabic — a language that is not one of those usually considered well represented on the Web — encourages us to believe that for other languages of interest we will be similarly successful. We have also done a bit of exploration to gauge the potential of the Archive for better represented language pairs, using English-Chinese as an example. By way of context, Chen and Nie (2000) reported that PTMiner found around 15,000 English-Chinese document pairs by crawling 185 sites in the `.hk` (Hong Kong) domain, with the run taking about a week. We did a STRAND search of the two Internet Archive crawls used in the English-Arabic study, seeking English-Chinese parallel text in multiple domains where Chinese is a dominant language (e.g. `.hk`, `.tw`, `.cn`). Our initial candidate pair set was generated in approximately 30 hours, and contains over 70,000 candidate page pairs. It is difficult to generalize from the English-Arabic experiment, but one cannot help but conclude that with even a fraction of the yield we obtained on the previous experiment, especially broadening out to include the other five sixths of the Archive, we should be able to obtain interestingly large quantities of English-Chinese text.

In terms of dissemination, the STRAND distribution mechanism models itself after Web search engines, distributing the URLs rather than the pages themselves. This places the legal burden on individual users, who are safe under "fair use" provisions if they download pages for their individual use. Until recently the difficulty with this solution has been that the collection of URLs deteriorates over time as sites disappear, pages are reorganized, and underly-

ing content changes — for example, in April 2002, we attempted to download the documents in the STRAND English-French, English-Chinese, and English-Basque collections, and we were able to successfully access only around 67%, 43%, and 40% of the URL pairs, respectively. However, the Internet Archive's Wayback Machine provides a way to distribute *persistent* URLs.

The quality of the data is a question that will need to be addressed by efforts to actually use those data in natural language applications. It is worth pointing out that, because STRAND expects pages to be very similar in structural terms, the resulting document collections are particularly amenable to sentence- or segment-level alignment. Indeed, just using dynamic programming to align the markup, ignoring the text, produces reasonable first-pass alignments of the intervening text as a side-effect. We are currently adapting statistical text-based sentence aligment techniques to take advantage of the markup available in Web-based document pairs.

Finally, our subjective impression is that parallel texts mined from the Web tend to contain a reasonable percentage of high quality translation, especially since pages often come from government or commercial sites. The underlying content, of course, is as rich and diverse as the Web itself, and what we as researchers can do with it is an exciting question that remains to be answered.

7 Acknowledgements

This work has been supported in part by Department of Defense Contract RD-02-5700, DARPA/ITO Cooperative Agreement N660010028910, and ONR MURI Contract FCPO.810548265. The second author is supported by a Fannie and John Hertz Foundation Fellowship. We would like to thank Nizar Habash, Kareem Darwish, Mona Diab, Ashraf Mohamed, and Usama Soltan for their assistance with Arabic, and Jason Eisner, Rebecca Hwa, and Doug Oard for helpful conversations. We would also like to thank Sang Hong for his assistance with implementation, and Brewster Kahle, Andy Jewell, Jad DeFanti, and Paula Keezer for permitting and facilitating our use of the Internet Archive.

References

Ahuja, Ravindra K., Thomas L. Magnati, and James B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, Englewood Cliffs, NJ.

Al-Onaizan, Yaser, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, I. Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation. Technical report, JHU. citeseer.nj.nec.com/al-onaizan99statistical.html.

Broder, Andrei Z., Steven C. Glassman, Mark S. Manasse, and Geoffery Zweig. 1997. Syntactic clustering of the web. In *Sixth International World-Wide Web Conference*, Santa Clara, CA, April.

Brown, P., J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Cabezas, Clara, Bonnie Dorr, and Philip Resnik. 2001. Spanish language processing at University of Maryland: Building infrastructure for multilingual applications. In *Proceedings of the Second International Workshop on Spanish Language Processing and Language Technologies (SLPLT-2)*.

Chen, Jiang and Jian-Yun Nie. 2000. Web parallel text mining for chinese english cross-language information retrieval. In *International Conference on Chinese Language Computing*, Chicago, Illinois.

Church, Kenneth W. and Robert Mercer. 1993. Introduction to the special issue on computational linguistics using large corpora. *Computational Linguistics*, 19(1):1–24.

Darwish, Kareem. 2002. Building a shallow arabic morphological analyser in one day. In *Workshop on Computational Approaches to Semitic Languages*, Philadelphia, July.

Davis, Mark and Ted Dunning. 1995. A TREC evaluation of query translation methods for multi-lingual text retrieval. In *Fourth Text Retrieval Conference (TREC-4)*. NIST.

Diab, Mona and Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In *40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, July.

Dunning, Ted. 1994. Statistical identification of language. Computing Research Laboratory Technical Memo MCCS 94-273, New Mexico State University, Las Cruces, New Mexico.

Gale, William A. and Kenneth W. Church. 1991. Identifying word correspondences in parallel texts. In *Fourth DARPA Workshop on Speech and Natural Language*, Asilomar, California, February.

Hunt, J. W. and M. D. McIlroy. 1975. An algorithm for differential file comparison. Technical Memorandum 75-1271-11, Bell Laboratories, October.

Hwa, Rebecca, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating translational correspondence using annotation projection. In *40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, July.

Landauer, Thomas K. and Michael L. Littman. 1990. Fully automatic cross-language document retrieval using latent semantic indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*, pages pages 31–38, UW Centre for the New OED and Text Research, Waterloo, Ontario, October.

Lin, Dekang. 1998. An information-theoretic definition of similarity. In *Fifteenth International Conference on Machine Learning*, Madison, WI, July.

Lopez, Adam, Michael Nossal, Rebecca Hwa, and Philip Resnik. 2002. Word-level alignment for multilingual resource acquisition. In *Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data, at the Third International Conference on Language Resources and Evaluation (LREC-2000)*, Las Palmas, Canary Islands, Spain, June.

Ma, Xiaoyi and Mark Liberman. 1999. Bits: A method for bilingual text search over the web. In *Machine Translation Summit VII*, September. http://www.ldc.upenn.edu/Papers/MTSVII1999/BITS.ps.

Melamed, I. Dan. 1997. Automatic discovery of non-compositional compounds in parallel data. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, Brown University, August.

Melamed, I. Dan. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, June.

Oard, Douglas W. 1997. Cross-language text retrieval research in the USA. In *Third DELOS Workshop*. European Research Consortium for Informatics and Mathematics, March.

Resnik, Philip. 1998. Parallel strands: A preliminary investigation into mining the Web for bilingual text. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas, AMTA-98, in Lecture Notes in Artificial Intelligence, 1529*, Langhorne, PA, October 28-31.

Resnik, Philip. 1999. Mining the Web for bilingual text. In *Proceedings of the 37th Annual Meeting of the ACL*, June.

Resnik, Philip and I. Dan Melamed. 1997. Semi-automatic acquisition of domain-specific translation lexicons. In *Fifth Conference on Applied Natural Language Processing*, Washington, D.C.

Resnik, Philip, Mari Broman Olsen, and Mona Diab. 1999. The Bible as a parallel corpus: Annotating the 'Book of 2000 Tongues'. *Computers and the Humanities*, 33:129–153.

Resnik, Philip and Noah A. Smith. 2002. Getting serious about "more is better": mining the internet archive for bilingual text. In review.

Riloff, Ellen, Charles Schafer, and David Yarowsky. 2002. Inducing information extraction systems for new languages via cross-language projection. In *Nineteenth International Conference on Computational Linguistics (COLING-2002)*, Taipei, Taiwan, August.

Smith, Noah A. 2001. Detection of translational equivalence. Undergraduate honors thesis, University of Maryland College Park. http://nlp.cs.jhu.edu/~nasmith/cmsc-thesis.ps.

Smith, Noah A. 2002. From matching words to matching corpora: recognizing translation. In review.

Tiedemann, Jörg. 1999. Automatic construction of weighted string similarity measures. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, College Park, MD, June.

Yarowsky, David and Grace Ngai. 2001. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Association for Computational Linguistics (NAACL-2001)*, Pittsburgh, PA, June.

Yarowsky, David, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research.*